View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Samara University

Литература

1 Белл, Д. Грядущее постиндустриальное общество [Текст] / Д.Белл. - М.: *Basic Books*, 2001. – 578 с.

2 Петров В. П., Петров С. В. Информационная безопасность человека и общества: учебное пособие. – М. : ЭНАС, 2007. – 336 с.

3 Конституция Российской Федерации. Принята всенародным голосованием 12 декабря 1993 года. С изм. 6-ФКЗ, 7-ФКЗ от [Текст] // Российская газета, 21.01.2009,. – Федеральный выпуск №7

4 Всеобщая декларация прав человека [Электронный ресурс] - Режим доступа: *http://constitution.garant.ru/act/right/megdunar/2540400/,* свободный - Яз., рус. – Загл с экрана.

5 Международный пакт о гражданских и политических правах [Электронный ресурс] - Режим доступа: *http://constitution.garant.ru/act/right/megdunar/2540400/*, свободный - Яз., рус. – Загл с экрана.

6 Конвенция о защите прав человека и основных свобод ETS N 005 [Электронный ресурс] - Режим доступа: *http://constitution.garant.ru/act/right/megdunar/2540400/*, свободный - Яз., рус. – Загл с экрана.

7 Хартия социальных прав и гарантий граждан независимых государств. [Электронный ресурс] - Режим доступа: *http://constitution.garant.ru/act/right/megdunar/2540400/*, свободный - Яз., рус. – Загл с экрана.

Д.В. Литвинов

ИССЛЕДОВАНИЕ ОПТИМАЛЬНОЙ СТРАТЕГИИ КЛАССИФИКАЦИИ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ГРАФА ПОТОКА ВЫПОЛНЕНИЯ

(Центр специальных разработок Министерства Обороны РФ)

Введение

Вредоносное ПО представляет серьезную угрозу для современных компьютерных систем. Все вредоносные программы можно разделить на семейства со схожей функциональностью. Целью классификации вирусов является как поиск новых семейств, так и определение принадлежности образца к уже существующему семейству.

В данной работе сравнивается эффективность двух стратегий классификации с целью поиска наиболее оптимального в смысле некоторого критерия. Под стратегией классификации в дальнейшем будем понимать способ определения расстояния между графами (с помощью марковских цепей и расстояния редактирования) и алгоритм кластеризации набора вредоносных программ на семейства (метод k средних и DBSCAN).

Выбор оптимальной стратегии классификации происходит по критериям минимума количества ошибок классификации программы на семейства и количества ошибок детектирования вредоносного ПО.

1. Построение графа потока управления

Сначала проведем статический анализ и построим граф вызовов функций, начав обход с точки входа в программу. Для каждой функции построим графы потока управления. Объединив граф вызовов функций и графы потоков управления функций получим граф потока управления программы [2]. Для статического анализа и построения графа потока управления используется Ida Pro Free Edition 5.0.

Статический анализ обладает хорошим покрытием кода, но он не может отслеживать переходы по адресам, которые вычисляются динамически во время выполнения программы. Избежать эту проблему помогает динамический анализ. Анализатор переходит на требуемые ветки графа, содержащие необработанные переходы, изменяя значения регистра RIP/EIP и регистра флагов процессора. Предполагается, что функции, на которые будет совершен переход, были найдены на этапе статического анализа, и по ним уже построен граф потока управления. Для динамического анализа используем инструментацию программы с помощью библиотеки PIN.

2. Построение классификаторов

В результате статического и динамического анализа программы был получен граф потока управления. Определим функции близости между графами *G* и *H* на основе расстояния редактирования и марковских цепей.

Для определения функции близости на основе расстояния редактирования будем сравнивать графы вызов функций, полученные из междпроцедурного графов потоков управления. Представим граф вызов функций в виде маркированного графа, где каждой вершине соответствует определенная функция. Расстояние редактирования между двумя графами G и H можно определить [3] как

$$\lambda_f(G,H) = VC(G,H) + FC(G,H) + EC(G,H),$$

где $VC(G,H)$ - число операций удаления и вставки вершин, $EC(G,H)$ - число несохранившихся ребер, $FC(G,H)$ - число несовпавших имен внешних функций. На практике за приемлемое время можно рассчитывать расстояния редактирования только для небольших графов (менее 50 вершин) [1]. Поскольку размеры графа потока управления программы существенно больше, попытаемся аппроксимировать расстояние редактирования с помощью алгоритма имитации отжига [4]. Определим *функцию близости* двух графов *G* и *H*:

$$\sigma_E(G,H) = \frac{\lambda_f(G,H)}{|V(G)| + |V(H)| + |E(G)| + |E(H)|}, \quad (1)$$

где $V(G)$ – множество вершин, $E(G)$ – множество ребер графа.

Построим марковскую цепь, узлами которой будут классы инструкций, а весом ребра - вероятность перехода между инструкциями, рассчитываемыми по графу потока управления. Пусть $A = \|a_{ij}\|$ - матрица, элементами которой яв-

ляются веса ребер полученного графа. Тогда определим *функцию близости* между двумя графами G и H как:

$$\sigma_M(G,H) = \frac{\Sigma_{i,j}\left(a_{ij}^G - a_{ij}^H\right)^2}{\Sigma_{i,j}\left(a_{ij}^G + a_{ij}^H\right)^2}, \quad (2)$$

3. Кластеризация на семейства

Для автоматического построения базы сигнатур семейств вредоносного ПО проведем кластеризацию программ из обучающей выборки по полученным графам потока управления по алгоритмам k средних и DBSCAN. В качестве расстояния между элементами будем использовать введенные ранее функции близости (1) и (2).

В алгоритме k-means мы предполагаем, что возможно провести кластеризацию и получить хорошо разделенные кластеры с четким центром. Но следует учитывать, что вредоносное ПО эволюционирует со временем. N-ая версия ПО будет похожа на n+1 версию, но совсем не обязательно, чтобы она была похоже на версии, далеко отстоящие от n-ой. Из этого следует что образцы вредоносного ПО совсем не обязательно будут образовывать кластеры, сферически располагающиеся вокруг одной точки. Потребуется отбрасывать образцы, для которых неясно к какому кластеру они принадлежат и рассматривать несферические кластеры. Это приводит нас к алгоритму пространственной кластеризации на основе плотности распределения с шумом DBSCAN [3].

В результате кластеризации была получена база сигнатур семейств вредоносного ПО. Элементы тестовой выборки классифицируются по методу ближайшего соседа на заранее полученные семейства вредоносного ПО.

Введем порог m для функций близости (1) и (2), при превышении которого программа не соотносится ни с одним кластером и будет соответствовать безопасному ПО.

4. Анализ результатов эксперимента

В основе тестовой выборки лежат 10 программ, написанных на C/C++ под ОС Windows 7 и скомпилированные под архитектуру x86. Все программы были протестированы на антивирусном ПО Kaspersky Internet Security 2013 и были им классифицированы как потенциально опасные. С помощью обфускации из каждой было получено по 20 программ. 100 из них составили обучающую выборку и 100 вошли тестовую. В качестве невредоносных программ для тестовой выборки были взяты 100 исполняемых файлов из системных папок ОС Windows.

В качестве параметров кластеризации возьмём [3]: MinPts=3, Rad=0,3, k=10. Точность классификации вредоносного ПО на семейства рассчитаем по формуле (как процент правильно классифицированных образцов ко всем обработанным):

$$P = \frac{1}{n}\sum_{c=1}^{n}\frac{a_{c,c}}{\Sigma_{i=1}^{n} a_{c,i}},$$

где a_{ij} обозначают число программ из семейства i отнесенных в кластер j.

Таблица 1

Алгоритм	P, %
K-means + MED	84,0
K-means + Markov chains	74,0
DBSCAN + MED	87,66
DBSCAN + Markov chains	78,27

Как видно из таблицы 1, наиболее точную классификацию дает алгоритм DBSCAN вместе с функцией близости по расстоянию редактирования (MED). При этом классификация по алгоритму k средних хуже, чем по алгоритму DBSCAN, а использование функции близости по марковской цепи дает худшие результаты, чем по функции близости по расстоянию редактирования.

На рисунке 1 представлена зависимость количества образцов вредоносного и безопасного ПО, взятых из тестовой выборки, от расстояния до ближайшего кластера для DBSCAN + MED. По оси абсцисс расположены значения функции близости до ближайшего кластера, по оси ординат – количество программ на данном расстоянии. Исходя из графиков, выберем оптимальное значение порога *m*, когда количество безопасного ПО будет превышать количество вредоносных ПО на заданном расстоянии.

На основании найденных порогов точность детектирования вредоносного ПО представлена в таблице 2 и рассчитана по формулам:

1) $P_{FN} = \frac{N_{FN}}{N} * 100\%$ – процент ошибок 2 рода;

2) $P_{FP} = \frac{N_{FP}}{N} * 100\%$ – процент ошибок 1 рода;

3) $P_T = \frac{N - N_{FN} - N_{FP}}{N} * 100\%$ – точность классификации,

Полученные результаты детектирования вредоносного ПО позволяют судить о том, что эволюционная модель развития ПО дает более точные результаты.

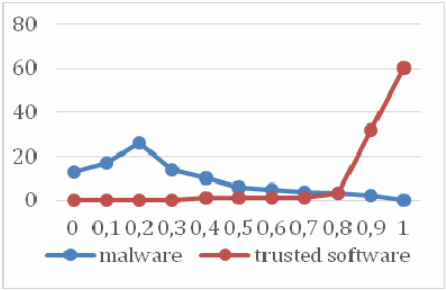

Рис. 1

Таблица 2

Алгоритм	P_T,%	P_{FP},%	P_{FN},%
K-means + MED	94,5	3,5	2,0
K-means + Markov chains	87,5	9,0	3,5
DBSCAN + MED	95,5	3,0	1,5
DBSCAN + Markov chains	90,0	7,5	2,5

Заключение

В данной работе исследованы две стратегии классификации вредоносного ПО по графу потока выполнения. Разработана методика построения графа потока выполнения, с помощью статический и динамический анализа. Рассмотрены два расстояния близости между графами (расстояние редактирования и на основе марковских цепей) и на их основе построены два алгоритма кластеризации: k-means и DBSCAN. Экспериментальные исследования показали, что модель эволюционного развития вредоносного ПО вместе с расстоянием близости графов, построенным на основе расстояния редактирования, является более оптимальной и дает лучшие результаты.

Литература

1. Large-scale malware indexing using function-call graphs / Hu, Xin, Tzicker Chiueh, and Kang G. Shin // In *Proceedings of the 16th ACM conference on Computer and communications security.* – 2009. - P. 611-620.

2. Efficient Virus Detection Using Dynamic Instruction Sequences / Dai, Jianyong, Ratan Guha, and Joohan Lee // *Journal of Computers*. – 2009. - Vol. 4, № 5.

3. Malware classification based on call graph clustering / Kinable, Joris, and Orestis Kostakis // *Journal in computer virology.* – 2011. – Vol. 7, №4. – P. 233-245.

4. Improved call graph comparison using simulated annealing / Kostakis, Orestis, Joris Kinable, Hamed Mahmoudi, and Kimmo Mustonen // In *Proceedings of the 2011 ACM Symposium on Applied Computing.* – 2011. – P. 1516-1523.

И.И. Набиев, И.М. Шаяхметов

ИССЛЕДОВАНИЕ ЭЛЕКТРОМАГНИТНЫХ ИЗЛУЧЕНИЙ ОТ ЭЛЕКТРОННЫХ СРЕДСТВ

(Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ)

Образование электромагнитного излучения от электронных средств (ЭС), связано с изменением тока или напряжения в электрических цепях при переключениях элементов [1]. Основной сложностью проведения экспериментальных измерений электромагнитного излучения от ЭС является необходимость в полубезэховой камере. Полубезэховая камера - экранированное помещение, внутренние поверхности которого покрыты поглощающим электромагнитные волны материалом, за исключением пола (пластины заземления), который должен отражать электромагнитные волны [2]. Данное оборудование имеет очень высокую цену и имеется в наличие у ограниченного количества организаций.

Целью данной работы является разработка простой экспериментальной методики и анализ электромагнитного излучения от ЭС на месте его эксплуатации. В качестве примера ЭВС используются персональные компьютеры.